Guidelines Overview

William L. Fithen, Software Engineering Institute [vita³]

Copyright © 2005 Carnegie Mellon University

2005-10-03

All systems have vulnerabilities, either in the technology from which they are constructed or in the behaviors of the people who use them.

Introduction

The Build Security In Guidelines is a taxonomy of mid-level engineering concerns that were derived from the vulnerability database accumulated by the CERT® Coordination Center over its 15-year history. In general, these concerns are less abstract than the Build Security In Principles⁵—which are intended to be enduring top-level concerns—and more abstract than the Build Security In Coding Rules⁶—which are intended to be precise, specific implementation advice.

The Taxonomy

- 1. Assume that Human Behavior Will Introduce Vulnerabilities into Your System⁸
- 2. Assume that Technology Will Contain Vulnerabilities
 - 1. Follow the Rules Regarding Concurrency Management⁹
 - 2. Design Configuration Subsystems Correctly and Distribute Safe Default Configurations¹⁰
 - 3. Carefully Study Other Systems Before Incorporating Them into Your System Through Delegation¹¹
 - 4. If Emulation of Another System Is Necessary, Ensure that It Is as Correct and Complete as Possible¹²
 - 5. Handle All Errors Safely¹³
- daisy:320 (Fithen, William L.) 3.
- 6. Validate All Input as Precisely as Possible daisy: 79 (Principles) 5.
- daisy: 16 Cooling Rules) 118 Properly Canonicalized 14 6.
- daisy 24-B(ASsumpi trian Humbon Be Favisti Nyil Untrodhen Multierab Fixture rind Roup (System) tion of Internal Data 8.
- daisy:332 (Full Wife Rules Regarding Concurrency Management)
- daisy:333Design Configuration Subsystems Storing (Bjand Distribute Safe Default Configurations)
- daisy:336 (Carefully Study Other Systems Before Incorporating Them into Your System)
 4. Treat the Entire Inherited Process Context as Unvalidated Input
- daisy:337 (If Emulation of Another System Is Necessary, Ensure that It Is as Correct and Complete as Possible)
- 5 Never Use Unvalidated Input as Part of a Directive to any Internal Component daisy:338 (Handle All Errors Safely)
- 14. daisy:331 (Ensure that Input Is Properly Canonicalized)
- 15. daisy:339 (Be Suspicious about Trusting Unauthenticated External Representation of Internal Data Structures)
- 16. daisy:340 (Do Not Use the "%n" Format String Specifier)
- 17. daisy:341 (Treat the Entire Inherited Process Context as Unvalidated Input)
- 18. daisy:342 (Never Use Unvalidated Input as Part of a Directive to any Internal Component)

Guidelines Overview

- 7. Use All Security Mechanisms Correctly
 - 1. Use Authentication Mechanisms, Where Appropriate, Correctly 19
 - 2. Use Authorization Mechanisms Correctly²⁰
 - 3. Use Well-Known Cryptography Appropriately and Correctly²¹
- 8. Do Not Allow Your System to Ever Use or Depend on Language Behaviors that Are "Undefined"
 - 1. Ensure that the Bounds of No Memory Region Are Violated²²
 - 2. Clear Discarded Storage that Contained Secrets and Do Not Read Uninitialized Storage²³
 - 3. Do Not Perform Arithmetic with Unvalidated Input²⁴

Description

In every phase of a system's development, under particular conditions, features added—or omitted—can introduce security vulnerabilities. To produce a safe and secure system, the competent, security-conscious engineer must

- learn the meaning of software assurance and be knowledgeable in the practice of supporting techniques,
- recognize the security implications of all functional requirements,
- recognize the security implications of missing requirements,
- recognize emergent behaviors in the system that have security implications,
- recognize the implications of an evolving deployment environment on the system,
- translate those implications into additional system requirements,
- design features to meet those requirements,
- recognize the security implications of the included and omitted features,
- add, modify, or remove features accordingly,
- recognize the security implications of the system's implementation,
- correct any defects in the implementation,
- understand how to test the system for compliance with security requirements, and
- be able to use software assurance techniques to demonstrate the assurance attributes of the system.

A failure in any of these, and more, can leave the system with security vulnerabilities.

19. daisy:321 (Use Authentication Mechanisms, Where Appropriate, Correctly)

20. daisy:322 (Use Authorization Mechanisms Correctly) **SEI Copyright**21. daisy:334 (Use Well-Known Cryptography Appropriately and Correctly)

Carrleisie 3 Mellow we think their Bostell afull holded a two considered so the U.S. Department of Defense undera Contract EA87221d055Cr000Bat Carnagia Melloun II niversity exetains conputing his gip all material produced under this contract. The U.S. Government retains a non-exclusive, royalty-free license to 24. daisy:343 (Do Not Perform Arithmetic with Unvalidated Input)

2

ID: 324 | Version: 6 | Date: 12/04/06 16:57:16

publish or reproduce these documents, or allow others to do so, for U.S. Government purposes only pursuant to the copyright license under the contract clause at 252.227-7013.

Permission to reproduce this document and to prepare derivative works from this document for internal use is granted, provided the copyright and "No Warranty" statements are included with all reproductions and derivative works.

For inquiries regarding reproducing this document or preparing derivative works of this document for external and commercial use, including information about "Fair Use," see the Permissions¹ page on the SEI web site. If you do not find the copyright information you need on this web site, please consult your legal counsel for advice.

Fields

Name	Value
Copyright Holder	SEI

Fields

Name	Value
is-content-area-overview	true
Content Areas	Knowledge/Guidelines
SDLC Relevance	Implementation
Workflow State	Publishable

ID: 324 | Version: 6 | Date: 12/04/06 16:57:16

 $^{1. \}quad http://www.sei.cmu.edu/about/legal-permissions.html\\$